ZJU Summer 2023 Contest 3

Contest Analysis

Group S

07.10.2023

A. Permutation Compression II

题目大意

- 给一个长度为 n 的排列，每次可以删除一个前缀最大值。求在删除次数最少的情况下使得是前缀最大值的位置最多，输出方案。

A. Permutation Compression II

题解

- 通过这样的操作可以让原序列的任何一个递增子序列成为答案。因为我们可以从前往后依次删除不在该子序列中的数。
- 那么前缀最大值的位置数量最多就是最长上升子序列。假设 LIS 为 $i_1, i_2, \ldots i_k$，那么对于所有 $i_t \le u \le i_{t+1}$，如果 $p_u \ge p_{i_t}$，我们就需要删除 p_u，所以对于一个 LIS 删除次数是固定的，等于符合上述条件的元素个数。
- 我们可以通过 DP 来求出最少有多少这样的元素，或者只需要贪心选择出现位置最靠前的 LIS，因为这样可以同时使得 p_{i_t} 尽量大，容易证明这样是正确的。
- 时间复杂度 $\Theta(n \log n)$。

- 给定 n, k, t，定义 $g(x) = k \cdot \phi(x)$。计算 $g^{(t)}(n)$.

B. Simple Calculation

题解

- 由于 φ 是积性函数，我们可以独立的考虑每个素因子。
- 每个 n 和 $g^{(\cdot)}(n)$ 中的素因子在 $2\log(n)$ 轮后就会消失。余下的 k 对于结果的影响 (在 $2\log(n)$ 轮之后) 是一个等比数列。
- 所以我们可以暴力模拟前 $\min(2\log(\max(n,k)),t)$ 轮，之后用快速幂计算剩余轮次的贡献。
- 使用 {std::map} 来维护素因子及其因子分解结果。
- 时间复杂度为 $\mathcal{O}(\sqrt{n}\log(n)+\sqrt{k}\log(k)+\log(t))$。

- 给一个序列，支持以下操作：
  - 单点修改。
  - 找出最小的 $i \in [l, r]$，使得将 A_i 替换成 v 之后原序列中相邻元素的大小关系不会改变。

- 对于 i, A_i 可以替换的值为一个区间。有四种情况:
  - $A_{i-1} \leq A_i \leq A_{i+1}$: $v \in [A_{i-1}, A_{i+1}]$.
  - $A_{i-1} \leq A_i > A_{i+1}$: $v \in [\max(A_{i-1}, A_{i+1}+1), \infty)$.
  - $A_{i-1} > A_i \leq A_{i+1}$: $v \in (-\infty, \min(A_{i-1}-1, A_{i+1})$.
  - $A_{i-1} > A_i > A_{i+1}$: $v \in [A_{i+1}+1, A_{i-1}-1]$.
- 令 $g(i)$ 为 A_i 对应的区间, 注意到对于区间 $[l, r]$, 对于区间内的 $g(i)$ 的并集为至多两个区间。用线段树维护这些区间，在线段树上二分，时间复杂度为 $\Theta(n \log n)$。

- 对于在区间 $[l, r]$ 中插入，我们只需要考虑前三个单调的段。
- 证明需要讨论所有情况，留作读者的练习。
- 可以用 `std::set` 维护所有满足 $A_{i-1} \le A_i > A_{i+1}$ 或 $A_{i-1} > A_i \le A_{i+1}$ 的 i，然后检查这些关键位置，并且在关键位置之间的序列二分即可。
- 时间复杂度也是 $\Theta(n \log n)$。

- 给定三个字符串集合 P, Q, R。询问有多少对 (A, B) 使得 A 是某个 P_i 的前缀，B 是某个 Q_j 的后缀且 AB 某个 R_k 的子串。

- 对 P 和 Q^R 分别建 AC 自动机，这里 Q^R 是翻转 Q 中的每个串。将他们称为 A_P, A_Q。
- 对于长为 l 的串 $S \in R$，假设 $pre(S,i), suf(S,i)$ 表示 $S_1S_2\dots S_i$ 和 $S_iS_{i+1}\dots S_l$.
- 对于某个 i, $pre(S,i)$ 在 A_P 中接受节点为 x 且 $suf(S,i+1)$ 在 A_Q 中接受节点为 y, 那么所有满足在 A_P 的 fail 树上 u 是 x 祖先，在 A_Q 的 fail 树上 v 是 y 的祖先的节点对 (u,v) 都代表一组合法的 (A,B)。

- 那么现在问题就是，给定两棵树和若干节点对 (x, y)，计算有多少对节点 (u, v) 使得存在 (x, y) ，满足在第一棵树上 u 是 x 的祖先，在第二棵树上 v 是 y 的祖先。
- 枚举节点 u，对于所有满足 x 在 u 子树内的 (x, y) 来说，可能的 v 即所有 y 到根的链的并中的节点。假设这个并集为 $S(u)$，那么我们可以通过合并所有 u 的儿子 t 的 $S(t)$ 来得到 $S(u)$。那么我们就可以用启发式合并或线段树合并维护这个集合和大小，时间复杂度 $\Theta(n \log n)$ 或 $\Theta(n \log^2 n)$。

E. Xor is Add

题目大意

- 构造一个排列 p, 使得 $p_i \oplus i = p_i + i$。

- 根据 $p_i + i = 2 \cdot (p_i \ \& \ i) + (p_i \oplus i)$，可以转化为 $p_i \ \& \ i = 0$。
- 考虑 $n-1$ 的最高位。假设为 k，那么我们可以将所有满足 $x \geq 2^k$ 的 x 与 $y = 2^{k+1} - 1 - x$ 配对，即 $p_x = y, p_y = x$。
- 然后问题的规模就被缩小为 $n' = 2^{k+1} - n$，并且边界情况为 $n = 1$ 或 $n = 0$，都是平凡的。
- 时间复杂度为 $\Theta(n)$。

- 给一张有向图，每条边可以以两种方法通过：
  - 使用 t 时间通过，有 $\frac{p}{100}$ 的概率被传送回 1 号点
  - 使用 c 时间通过
- 计算从 1 到 n 的期望最小时间。

- 当我们处在同一节点时，最优策略总是固定的，如果我们被传送回 1 号节点，那么我们一定会沿着相同的路径回来。
- 我们可以用 Dijkstra 算法来计算最短路，时间复杂度 $\Theta(m\log n)$。
- 我们也可以二分答案，从 n 号点倒推，但是需要注意实现的常数，时间复杂度为 $\Theta(m\log n\log(\frac{n\cdot maxw}{\varepsilon}))$。

- T 次询问，每次给两个圆 $(x_1, y_1, r_1), (x_2, y_2, r_2)$，询问在第一个圆上有多少长度使得顺时针方向切线和第二个圆相交，且距离 $\leq L$。

G. Open Trains

题解

- 求出两个圆的外公切线和内公切线，一定是夹在一条外公切线和内功切线中间的一段。
- 注意到切线到第二个圆的长度在这段上先递减再递增，可以三分找到最小值再在两边二分求出答案，时间复杂度 $\Theta(T\log(\frac{1}{\varepsilon}))$。
- 也可以用几何方法直接得到表达式计算，时间复杂度 $\Theta(T)$。

Thanks!